

CLAIMS

1. A method for texture compressing images having a plurality of color components (R, G, B), including decomposing the images in sub-blocks each including only 5 one said color component (R, G, B), the method comprising:
defining at least one first predictor (R₂₂, G₂₂, B₂₂) for each said sub-block;
computing for each said sub-block a respective set of prediction differences between at least some of the 10 pixels of the sub-block and said predictor (R₂₂, G₂₂, B₂₂);
sorting the prediction differences for each sub-block;
generating a look-up prediction differences palette 15 and defining therefrom a look-up prediction error palette; and
associating a code with each column of said error palette.

2. The method of claim 1, wherein said code 20 associated with each column of said error palette comprises a two-bit code (00, 01, 10, 11).

3. The method of claim 1, further comprising:
computing for each pixel (P_{ij}) in said images a respective prediction error (E_{ij}) using as a predictor a 25 given pixel (P₂₂) of said images, said given pixel (P₂₂) being the union of the first predictors for said sub-blocks (R, G, B);
for each said prediction error (E_{ij}), computing the Euclidean distance (Dist_j) between said prediction error 30 and each look-up color in said look-up prediction

palette, the difference being between homologous prediction error components;

for each respective prediction error (E_{ij}) finding the minimum of said distance; and

5 composing a bitstream including P22 used as a predictor and Min_error and Max_error used as the minimum and maximum prediction errors per each R,G,B component in the block under coding.

4. The method of claim 1, further comprising
10 defining said first predictor (R22, G22, B22) for each said sub-block (R, G, B) as a prediction for the respective sub-block excluding the predictor (R22, G22, B22) itself.

5. The method of claim 1, further comprising
15 computing, for each said sub-block, a respective set of prediction differences comprised of the differences of said predictor to all the other pixels in the respective sub-block (R, G, B).

6. The method of claim 1, further comprising sorting
20 said prediction differences in ascending order in each said sub-block (R, G, B).

7. The method of claim 1, wherein generating said look-up prediction differences palette comprises:

generating a first value comprising the minimum of
25 the prediction differences of each said sub-block (R, G, B), notated min_errorR, min_errorG, and min_errorB, respectively;

generating a second value comprising the maximum of
30 the prediction differences of each said sub-block (R, G, B), notated max_errorR, max_errorG, and max_errorB, respectively;

generating a third value, notated Int1, comprising
(a*min_errorR+b*max_errorR)/(a+b),
(c*min_errorG+d*max_errorG)/(c+d),
(e*min_errorB+f*max_errorB)/(e+f);
5 and generating a fourth value, notated Int2,
comprising
(g*min_errorR+h*max_errorR)/(g+h),
(i*min_errorG+l*max_errorG)/(i+l),
(m*min_errorB+n*max_errorB)/(m+n),
10 wherein a, b, c, d, e, f, g, h, i, l, m, and n are
weighing factors.

8. The method of claim 7, wherein generating said
look-up prediction error palette further comprises
generating an error palette having the following format:

15 [Min_errorR, Int1R, Int2R, Max_errorR]
[Min_errorG, Int1G, Int2G, Max_errorG]
[Min_errorB, Int1B, Int2B, Max_errorB].

9. The method of claim 8, further comprising
associating a predetermined two-bit code to each column
20 of said look-up prediction error palette.

10. The method of claim 7, further comprising
computing Euclidean distance values wherein:

Dist1 = $\sqrt{(|E_{Rij}-\text{Min_errorR}|^2 + |E_{Gij}-\text{Min_errorG}|^2 + |E_{Bij}-\text{Min_errorB}|^2)}$
Dist2 = $\sqrt{(|E_{Rij}-\text{Int1R}|^2 + |E_{Gij}-\text{Int1G}|^2 + |E_{Bij}-\text{Int1B}|^2)}$
25 Dist3 = $\sqrt{(|E_{Rij}-\text{Int2R}|^2 + |E_{Gij}-\text{Int2G}|^2 + |E_{Bij}-\text{Int2B}|^2)}$
Dist4 = $\sqrt{(|E_{Rij}-\text{Max_errorR}|^2 + |E_{Gij}-\text{Max_errorG}|^2 + |E_{Bij}-\text{Max_errorB}|^2)}.$

11. The method of claim 1, further comprising
defining said first predictor (R22, G22, B22) for each
sub-block as a prediction for a set of colors surrounding
30 said predictor (R22, G22, B22).

12. The method of claim 11, further comprising:
defining two groups of said sorted prediction differences, said groups being composed of the lowest elements and the highest elements in the sorting; and
5 generating said look-up prediction differences palette wherein,
a first value comprises the median of said first group of prediction differences for each said sub-block, notated `min_median_errorR`, `min_median_errorG`, and
10 `min_median_errorB`, respectively,
a second value comprises the median of said second group of prediction differences for each sub-block, notated `max_median_errorR`, `max_median_errorG`,
`max_median_errorB`, respectively,
15 a third value, notated `Int1`, comprises
$$(a*min_median_errorR+b*max_median_errorR)/(a+b),$$
$$(c*min_median_errorG+d*max_median_errorG)/(c+d),$$
$$(e*min_median_errorB+f*max_median_errorB)/(e+f),$$
and a fourth value, notated `Int2`, comprises
20
$$(g*min_median_errorR+h*max_median_errorR)/(g+h),$$
$$(i*min_median_errorG+l*max_median_errorG)/(i+l),$$
$$(m*min_median_errorB+n*max_median_errorB)/(m+n)$$
wherein `a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`, `i`, `l`, `m`, and `n` are weighing factors.
25
13. The method of claim 12, wherein generating said look-up prediction error further comprises generating an error palette having the following format:
[`Min_median_errorR`, `Int1R`, `Int2R`, `Max_median_errorR`]
[`Min_median_errorG`, `Int1G`, `Int2G`, `Max_median_errorG`]
30 [`Min_median_errorB`, `Int1B`, `Int2B`, `Max_median_errorB`].

14. The method of claim 13, further comprising associating a predetermined code to each column of said error palette.

15. The method of claim 1, further comprising:
5 defining a second set of predictors (R23, R22, R32, R33; G22, G23, G32, G33; B22, B23, B32, B33) and a plurality of respective scanning patterns in each sub-block (R, G, B) to generate a prediction error among said predictors and the pixel of the block;

10 computing said respective prediction error (E_{ij}), thus generating a plurality of sets of prediction differences for said sub-blocks, whereby a corresponding plurality of coding options are available corresponding to said sets, said plurality being determined by said 15 sets and the number of said sub-blocks;

coding said image by means of said plurality of coding options;

20 computing, for each image block and each color component (R, G, B), a difference value with respect to the source colors for all colors inside the block before encoding; and

finding the minimum for said value over said plurality of coding options.

16. The method of claim 15, further comprising
25 computing said difference value as a sum of squared differences.

17. The method of claims 15 further comprising:
composing a bitstream including a P22 pixel;
associating said code to each pixel in said image
30 other than said P22 pixel, whereby one of said plurality of options is coded;

adding a group of additional bits to the bitstream;
coding one virtual bit;
sending said bitstream to a decoder process to
generate decoded colors; and

5 computing said difference value before encoding for
each image and for each color component (R, G, B).

18.. The method of claim 1, wherein said images
comprise RGB color images and said color components
comprise the R, G, and B components of said RGB images.

10 19. The method of claim 3, further comprising:
deriving P22, P22 + Min_error, and P22 + Max_error
from said bitstream, and obtaining therefrom Min_error
and Max_error;

computing said look-up prediction error palette
15 using said bit code to address said look-up table; and
adding the value thus retrieved to P22 to recreate
each pixel in said images (P_{ij}).

20. The method of claim 17, further comprising:
deriving P22 from said bitstream; and
computing said look-up prediction error palette by
using said code to address said look-up table and adding
the value stored at this address to P22 to recreate each
pixel (P_{ij}), using the colors thus decoded as predictors,
and using said code to address said look-up table and
25 adding the value stored at this address to said
predictors to recreate each remaining color.

21. A processor for texture compressing images
having a plurality of color components (R, G, B),
including decomposing the images in sub-blocks each
30 including only one said color component (R, G, B),
comprising:

means for defining at least one first predictor (R22, G22, B22) for each said sub-block;

5 means for computing for each said sub-block a respective set of prediction differences between at least some of the pixels of the sub-block and said predictor (R22, G22, B22);

means for sorting the prediction differences for each sub-block;

10 means for generating a look-up prediction differences palette and defining therefrom a look-up prediction error palette; and

means for associating a code with each column of said error palette.

22. The processor of claim 21 wherein said processor
15 comprises a dedicated processor.

23. The processor of claim 21 wherein said processor comprises a programmed general-purpose processor.

24. A computer program product directly loadable into the memory of a digital computer and including
20 software code portions for performing the following method when the product is run on a computer, the method comprising:

defining at least one first predictor (R22, G22, B22) for each said sub-block;

25 computing for each said sub-block a respective set of prediction differences between at least some of the pixels of the sub-block and said predictor (R22, G22, B22);

30 sorting the prediction differences for each sub-block;

generating a look-up prediction differences palette
and defining therefrom a look-up prediction error
palette; and
associating a code with each column of said error
5 palette.